

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**MĚŘENÍ POLOHY OBJEKTU V PROSTORU POMOCÍ
DVOJICE KAMER**

MEASURING POSITIONS OF OBJECTS USING A PAIR OF CAMERAS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Adam Sláma

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Kříž

BRNO 2017



Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Adam Sláma

ID: 174393

Ročník: 3

Akademický rok: 2016/17

NÁZEV TÉMATU:

Měření polohy objektu v prostoru pomocí dvojice kamer

POKYNY PRO VYPRACOVÁNÍ:

S využitím jazyka C++ a knihoven OpenCV vytvořte funkční aplikaci, která bude demonstrovat použití Vámi realizovaného měřicího zařízení tvořeného dvojicí kamer a algoritmy extrahující prostorovou informaci z digitálních obrazů. Aplikace bude umožňovat určit 3D souřadnice zvolených objektů ve snímané scéně a to se zjištěnou přesností vzhledem k referenčnímu souřadnicovému systému.

DOPORUČENÁ LITERATURA:

[1] BRADSKI, G. Learning OpenCV: Computer Vision in C with the OpenCV Library. O'Reilly Media, 2012. ISBN 978-144-9314-651.

[2] HARTLEY, R. I., ZISSERMAN, A. Multiple View Geometry in Computer Vision. Second Edition. Cambridge University Press, 2004. 670 s. ISBN 9780521540513.

Termín zadání: 1.2.2017

Termín odevzdání: 8.6.2017

Vedoucí práce: Ing. Petr Kříž

Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce se zaměřuje na získávání prostorové informace z digitálního obrazu. Vzniklo jednoduché měřicí zařízení ze dvou statických kamer a vytvořeného softwaru, které dokáže s určitou přesností najít souřadnice měřeného objektu ve snímaném prostoru vzhledem k referenčnímu souřadnicovému systému. Program byl tvořen ve vývojovém prostředí Microsoft Visual Studio 2015 s balíčkem knihoven OpenCV v programovacím jazyce C++. Program je založen na kalibraci obrazu, určení světových a pixelových souřadnic a metodě triangulace.

KLÍČOVÁ SLOVA

OpenCV, zkreslení kamery, kalibrace kamery, triangulace, kamera Drift, C++

ABSTRACT

The thesis focuses on acquiring spatial information of a digital image. Simple measuring device consists of two static cameras and of a created software, which is able to find the coordinates of the measured object in the sensing area with certain accuracy due to the reference coordinate system. The program was formed in the development environment Microsoft Visual Studio 2015 using package of OpenCV libraries in C++ programming language. The program is based on calibration of image, determination of the global and pixel coordinates and on triangulation method.

KEYWORDS

OpenCV, camera distortion, camera calibration, triangulate, camera Drift, C++

SLÁMA, Adam *Měření polohy objektu v prostoru pomocí dvojice kamer*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 44 s. Vedoucí práce byl Ing. Petr Kříž

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Měření polohy objektu v prostoru pomocí dvojice kamer“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Petru Křížovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora

OBSAH

Úvod	10
1 Vývojové prostředí	11
1.1 Microsoft Visual Studio	11
1.2 OpenCV	11
2 Popis scény a parametry kamer	13
2.1 Scéna	13
2.2 Kamery Drift	13
2.2.1 Parametry pořízeného obrazu	13
3 Zkreslení a kalibrace obrazu	15
3.1 Zkreslení	16
3.2 Kalibrace	16
4 Triangulace	18
4.1 Lineární triangulace	18
4.2 Pravděpodobnost rozložení odhadovaného 3D bodu	19
5 Nejistoty měření	21
5.1 Odchytky měření	21
6 Implementace metod	23
6.1 Kalibrační snímky	24
6.2 Kalibrace kamery Drift	25
6.3 Pixelové souřadnice	27
6.4 Světové souřadnice	28
6.5 Pozice kamer vůči světovému souřadnicovému systému	28
6.6 Projekční matice	29
6.7 Nalezení bodu v prostoru	29
6.7.1 Předpřípravení obrazu	29
6.7.2 Nalezení bodu a výpis souřadnic	30
6.8 Popis chodu programu	31
7 Zhodnocení výstupních hodnot	33
7.1 První měření – kamery svírající nulový úhel	34
7.2 Druhé měření – kamery svírající úhel 40°	35
7.3 Třetí měření – kamery svírající úhel 85°	37

8 Závěr	39
Literatura	40
Seznam symbolů, veličin a zkratk	41
Seznam příloh	42
A Obsah přiloženého CD	43

SEZNAM OBRÁZKŮ

1.1	Logo Microsoft Visual Studio.	11
1.2	Logo knihovny OpenCV.	12
2.1	Pohled zepředu (a), pohled shora (b).	13
3.1	Paralelní promítání.[5]	15
3.2	Perspetivní promítání.[5]	15
3.3	Nezkreslený obraz (a), soudkovité zkreslení (b), poduškovité zkreslení (c).	16
3.4	Šachovnice pro kalibraci kamery.	17
4.1	Znároznění triangulace.	18
4.2	Nejistota rekonstrukce bodu.	20
6.1	Diagram řešení.	23
6.2	Snímek šachovnice pořízený kamerou Drift určený ke kalibraci.	24
6.3	Kontrolní vykreslení korespondenčních rohů.	25
6.4	Zkreslený snímek pořízený kamerou Drift.	26
6.5	Snímek s odstraněným zkreslením s kalibrační chybou 7,00291.	26
6.6	Snímek s odstraněným zkreslením s kalibrační chybou 1,8.	27
6.7	Snímek s odstraněným zkreslením s kalibrační chybou 0,561422.	27
6.8	Screenshot okna se souřadnicemi objektu v daném snímku.	32
7.1	Fotografie prvního měření (kamery svírají nulový úhel).	33
7.2	Fotografie druhého měření (kamery svírají úhel 40°).	34
7.3	Fotografie třetího měření (kamery svírají úhel 85°).	34
7.4	Prostor ve kterém měření probíhalo a číselně označené pozice.	35

SEZNAM TABULEK

2.1	Technické parametry kamery Drift Ghost-S.[4]	14
7.1	Reálné hodnoty měřeného objektu v daných pozicích.	36
7.2	Naměřené souřadnice objektu v daných pozicích v prvním měření.	36
7.3	Naměřené souřadnice objektu v daných pozicích ve druhém měření.	37
7.4	Naměřené souřadnice objektu v daných pozicích ve třetím měření.	38
7.5	Vypočtené hodnoty.	38

ÚVOD

Tato bakalářská práce se zabývá získáváním prostorové informace objektu v digitálních obrazech pomocí triangulační metody. Ze dvou statických kamer a vytvořeného softwaru (ve vývojovém prostředí Microsoft Visual Studio 2015 s balíčkem knihoven OpenCV v programovacím jazyce C++) vzniklo měřicí zařízení, které dokáže s určitou přesností zjistit 3D souřadnice zvoleného objektu ve snímané scéně vzhledem k referenčním souřadnicovému systému.

Teoretická část se skládá z pěti kapitol. V první kapitole s názvem Vývojové prostředí je stručně popsán program Microsoft Visual Studio a do něj implementované knihovny OpenCV. Ve druhé kapitole se nachází informace o snímané scéně a kamerách firmy Drift, které danou scénu snímají. Další kapitola pojednává o zkreslení obrazu a možnostech jeho odstranění (kalibrace). Ve čtvrté kapitole je přiblížen princip lineární triangulace. A pátá kapitola je zaměřena na odchylky měření, které byly ve výsledku spočítány.

Následující kapitoly jsou součástí praktické části bakalářské práce. Je zde popsán postup zpracování kalibračních snímků za účelem získání intrinsických parametrů kamer – kalibrace, dále jsou zde vysvětleny pojmy jako pixelové souřadnice, světové souřadnice, pozice kamer vůči světovému souřadnicovému systému, projekční matice a samotné nalezení objektu v prostu. Následně je věnována pozornost samotnému programu, jeho ovládání a vyhodnocení výstupních hodnot, které byly získány ze tří měření. Každé měření mělo jiné rozestavění kamer, ale hledaný objekt byl měřen na stejných pozicích.

V závěru je celkově hodnocena funkčnost celého projektu a možné využití v praxi.

1 VÝVOJOVÉ PROSTŘEDÍ

Jak už bylo výše zmíněno, k vypracování bakalářské práce bude využito vývojové prostředí Microsoft Visual Studio 2015 doplněné o balíček knihoven OpenCV.

Projekt bude psán v programovacím jazyce C++, který vznikl rozšířením jazyka C. Jedná se o jazyk, který podporuje procedurální, objektově orientované a generické programování. Díky tomu patří C++ v současné době mezi nejrozšířenější programovací jazyky.

1.1 Microsoft Visual Studio

Microsoft Visual Studio od společnosti Microsoft Corporation je jedním z nejpoužívanějších programovacích softwarů. Můžete zde tvořit a testovat desktopové aplikace, aplikace pro web či zařízení. V jednom IDE můžete psát kód, jak pro iOS, Android, tak i Windows, a zároveň ho můžete provozovat na různých platformách OS Windows a dalších zařízeních. Díky snadné navigaci, přehlednosti, IntelliSense atd. zvýší program vaši produktivitu a výkon.

Microsoft Visual Studio obsahuje integrovaný debugger, který pracuje na úrovni kódu i stroje, editor kódu, designer formulářů pro tvorbu aplikací GUI, designer webu, tříd a databázových schémat. Editor kódu a debugger podporují jakýkoliv programovací jazyk prostřednictvím jazykových služeb. Vestavěnými jazyky jsou C/C++, VB.NET a C#. Další jazyky pomocí jazykové služby musí být nainstalovány samostatně. Patří sem Oxygene, F#, Python a Ruby atd.. Dále je podporován JavaScript, XML/XSLT, HTML/XHTML a CSS. [1]



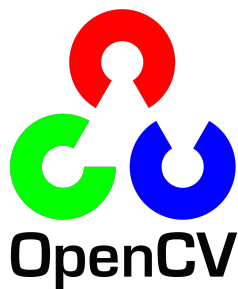
Obr. 1.1: Logo Microsoft Visual Studio.

1.2 OpenCV

OpenCV (Open Source Computer Vision) je otevřená knihovna počítačového vidění a strojového učení především v reálném čase, která je nativně psaná v jazyce

C++. Knihovna má licenci BSD, a proto je zdarma jak pro akademické účely, tak i pro komerční využití. [2]

Knihovna má C ++, C, Python, Java a MATLAB rozhraní a podporuje Windows, Linux, Android a Mac OS. Má ji více jak 47 tisíc uživatelů a odhadovaný počet stažení je přes 7 miliónů. Knihovna obsahuje více než 500 funkcí. Používaná je především ve firmách, výzkumných skupinách a vládních orgánech. Využití knihovny je velmi členité, od triviálních úkolů až po pokročilé, například v robotice. [3]



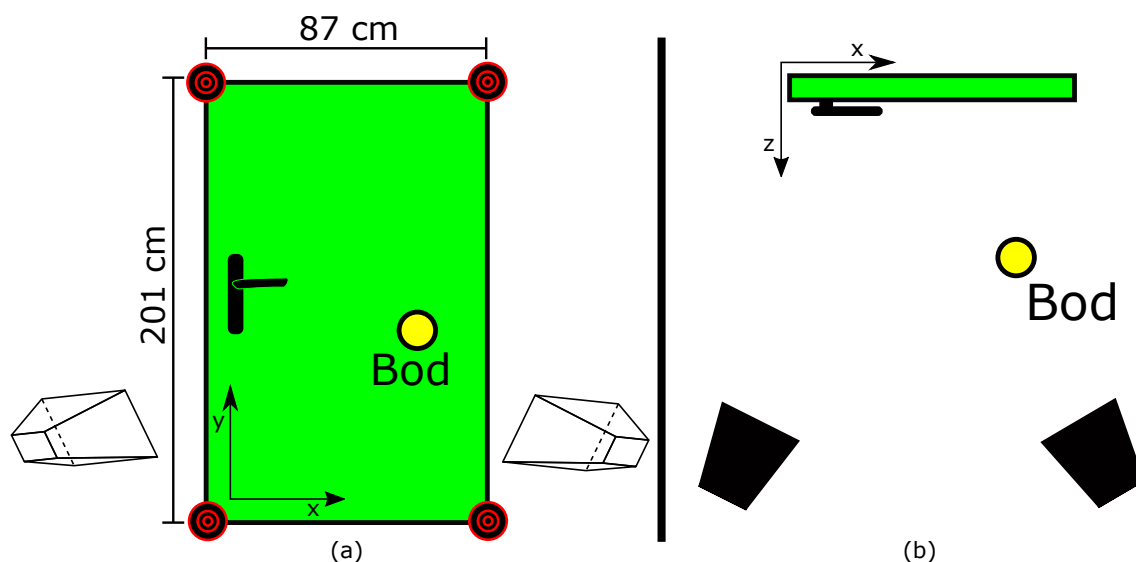
Obr. 1.2: Logo knihovny OpenCV.

2 POPIS SCÉNY A PARAMETRY KAMER

Námi zvolenou scénou se stal prostor před dveřmi, který byl natočen pomocí dvou sportovních vysokorychlostních kamer značky **Drift**.

2.1 Scéna

Pro zaznamenání a následné zpracování obrazu jsme si jako scénu vybrali prostor před dveřmi, kde rohy zárubně mají být reálné body v rovině (x, y) . Jako počátek kartézského systému vždy definujeme levý spodní roh. Aby byl bod k nalezení a zároveň byly určeny jeho souřadnice $[X, Y, Z]$, musí se nacházet na obraze obou kamer. Jednotlivé vzdálenosti a umístění v prostoru je znázorněno na obrázku 2.1.



Obr. 2.1: Pohled zepředu (a), pohled shora (b).

2.2 Kamery Drift

Důležité parametry použitých kamer značky Drift jsou uvedeny v tabulce 2.1.

Velký úhel záběru 160° nám zajišťuje dostatečně velký prostor, ve kterém bude program následně určovat souřadnice námi zvoleného bodu v prostoru. Nejzásadnější je pro nás kvalita pořízeného snímku.

2.2.1 Parametry pořízeného obrazu

Barevný záznam scény byl natočen v rozlišení 848×480 ve formátu *.mp4*. Snímková rychlost byla 25 fps. Program použije všechny snímky záznamu. U těch, na kterých

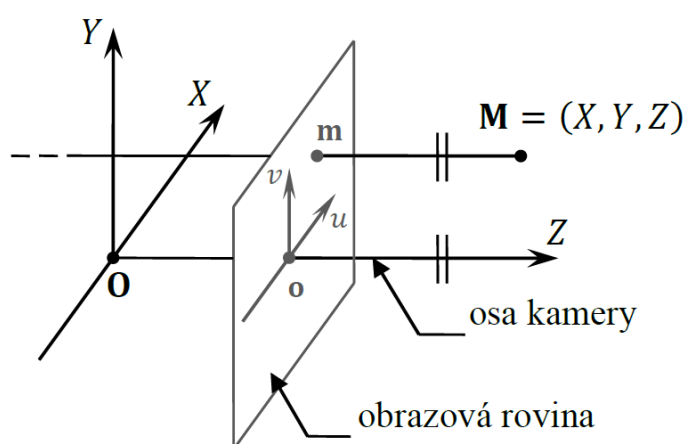
Tab. 2.1: Technické parametry kamery Drift Ghost-S.[4]

Parametr	Hodonota
Video rozlišení	1080p, 960p (4:3), 720p a WVGA (16:9)
Formát souboru	.MP4 (H.264 kodek)
Typ snímáče	CMOS
Ohnisková vzdálenost	0,4
Úhel záběru	160°
Snímková frekvence	max. 240 fps (WVGA)

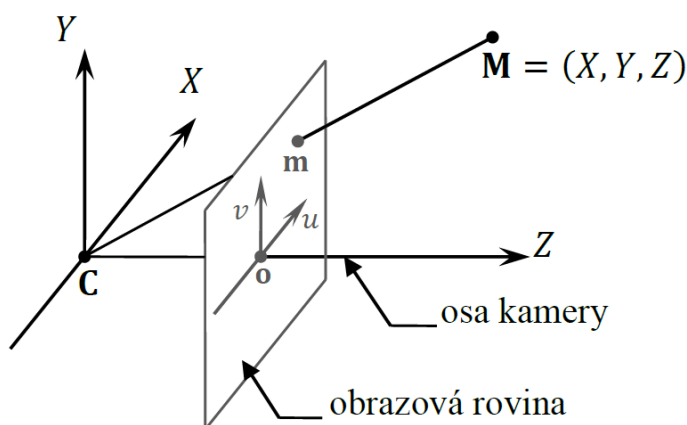
najde předem definovaný objekt, vypočítá jeho 3D souřadnice.

3 ZKRESLENÍ A KALIBRACE OBRAZU

Slovo „kamera“ můžeme chápat jako mapování souřadnic trojrozměrného prostoru (tzv. světových souřadnic) do dvourozměrného obrazového prostoru (tzv. obrazové roviny) viz obrázky 3.1 a 3.2. Pomocí maticového zápisu lze vyjádřit jednotlivé vlastnosti kamery realizující tuto funkci.[5]



Obr. 3.1: Paralelní promítání.[5]



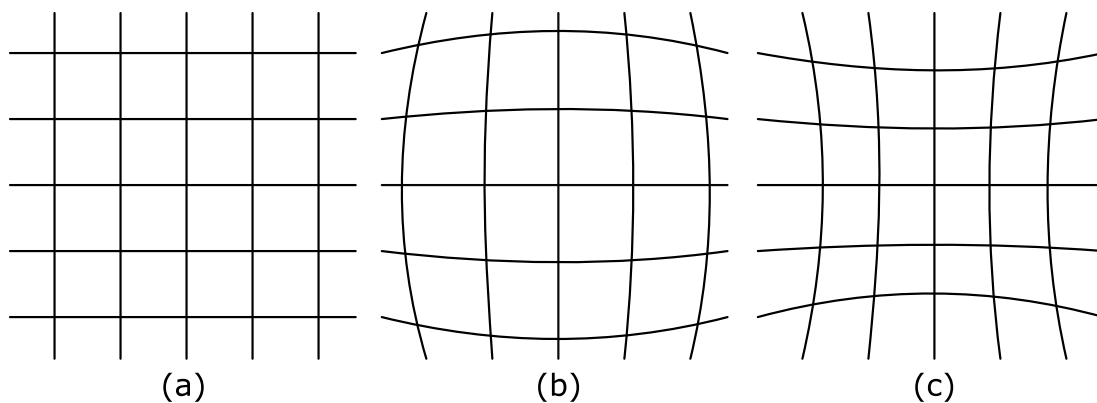
Obr. 3.2: Perspektivní promítání.[5]

Paralelní mapování se používá hlavně v systémech CAD, kde se snímané souřadnice prostoru rovnají obrazovým souřadnicím, protože mapovací přímka dopadá kolmo jak na obrazovou rovinu (u, v) , tak na projekční rovinu (X, Y) , tzn. $X = u$ a $Y = v$, tak jak je to znázorněno na obrázku 3.1. Souřadnice hloubky Z je zde ignorována.

Perspektivní mapování (středové) je takové, kde mapovací přímka prochází středem projekce C a bodem, který je promítán. S tímto promítáním se setkáme u většiny kamer a patří mezi ně i naše.

3.1 Zkreslení

Kvůli nedokonalosti objektivů se nám v informacích získaných pomocí kamer objevují optické chyby tzv. radiální zkreslení. Tyto chyby se nejčastěji vyskytují u kamer s širokoúhlým objektivem, tak jako tomu je i u naší kamery Drift. Rozdělují se na několik typů – soudkovitost, poduškovitost (viz obrázek 3.3), komplexní zkreslení, atd.. [6]

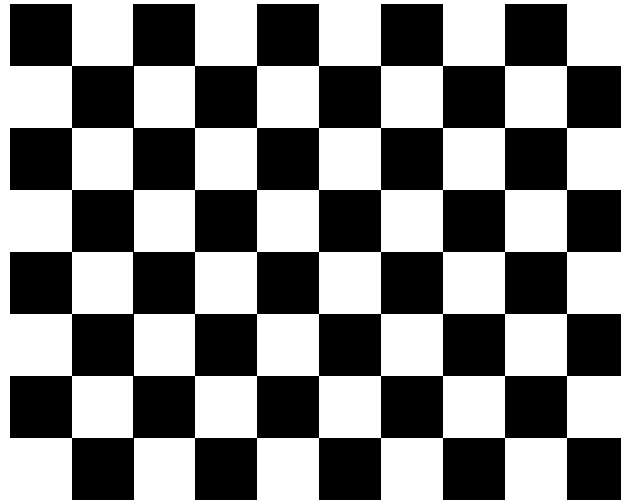


Obr. 3.3: Nezkreslený obraz (a), soudkovité zkreslení (b), poduškovité zkreslení (c).

3.2 Kalibrace

Hlavní funkce kalibrace kamery je odstranění nedostatků kamerového optoelektronického systému způsobených tangenciálním či radiálním zkreslením čočky nebo parametry CMOS senzoru, kterými je například nepravý úhel mezi osami senzoru, nebo jiný než čtvercový rozměr pixelu. Matematický model, kterým je kamera popsána, zahrnuje vnější (poloha a natočení kamery) a vnitřní (ohnisková vzdálenost, pozice hlavního bodu, zkreslení digitalizace a výše zmíněné tangenciální a radiální zkreslení) parametry. Výsledkem kalibrace je informace definující vztah mezi souřadnicemi snímaného procesu a souřadnicemi v projektivní rovině. K tomu se používá například fotometrická kalibrace, kde je nasnímán standardní kalibrační vzor (obrázek 3.4), u které jsou známy vlastnosti (u šachovnice počet čtverců), pod různou prostorovou orientací. Z pořízených snímků jsou poté automaticky nalezeny rohy obrazců, kde potom při porovnání očekávaných a následně zjištěných pozic dojde

k určení vnitřních parametrů. S větším počtem správných kalibračních snímků roste také přesnost měření.



Obr. 3.4: Šachovnice pro kalibraci kamery.

Kalibrační matici \mathbf{K} získáme součinem extrinsické \mathbf{G} a intrinsické \mathbf{P} , tedy

$$\mathbf{K} = \mathbf{P}\mathbf{G}, \quad (3.1)$$

kde

$$\mathbf{P} = \mathbf{A}\mathbf{P}_0. \quad (3.2)$$

Tzn. matice \mathbf{K} slučuje matici standardní normalizované perspektivní projekce \mathbf{P}_0 a matici vnitřních parametrů kamery \mathbf{A} , kdy

$$\mathbf{A} = \begin{pmatrix} a_u & 0 & u_0 \\ 0 & a_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (3.3)$$

zvětšení v obou osách určuje a_u a a_v a polohu počátku obrazové roviny definuje u_0 a v_0 ,

$$\mathbf{P}_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (3.4)$$

$$\mathbf{G} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3^T & 1 \end{pmatrix}, \quad (3.5)$$

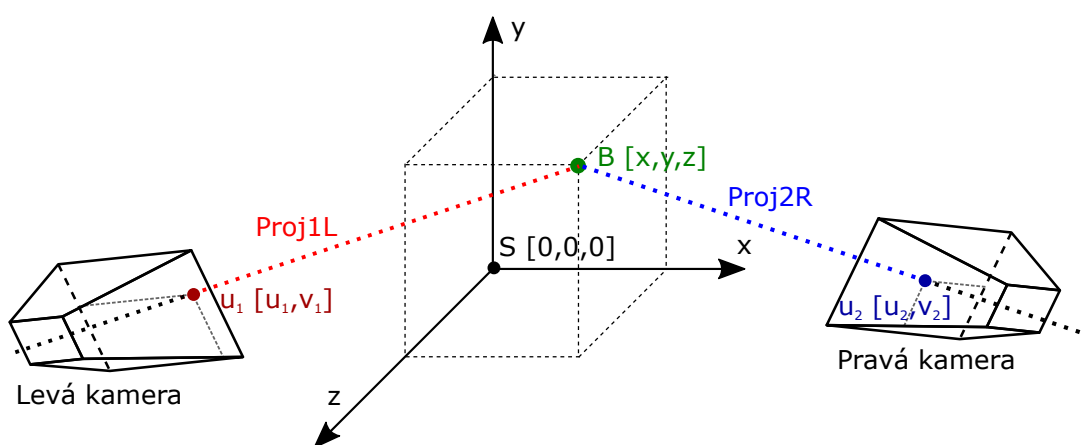
kde rotaci kamery vůči světovému souřadnicovému systému udává \mathbf{R} a \mathbf{t} je vektor, který určuje posunutí středu promítání oproti počátku. [7]

4 TRIANGULACE

Triangulace má více variant, které se nám sice zdají odlišné, ale pracují na stejných principech (trigonometrické výpočty). Je to metoda, která by nám měla pomoci určit s větší či menší přesností (záleží nejen na zvoleném typu triangulace) polohu bodu v 3D prostoru ze snímků pořízených kamerami.

V této práci se zaměříme na triangulace lineární.

4.1 Lineární triangulace



Obr. 4.1: Znázornění triangulace.

Mezi nejpoužívanější a výpočetně nejméně náročné triangulační metody patří lineární triangulace, která je zároveň implementována jako funkce OpenCV.

Představme si, že máme 3D prostor se zvoleným referenčním souřadnicovým systémem s počátkem $\mathbf{S} = (0, 0, 0)^T$. V prostoru se nachází bod $\mathbf{B} = (x, y, z)^T$ a dvě kamery, které daný bod zabírají. Tzv. projekční matice **Proj1L** a **Proj2R** získáme kalibrací, z čehož plyne

$$\mathbf{Proj1L} = \mathbf{A}_1 \mathbf{P}_0 \mathbf{G}_1, \quad (4.1)$$

přičemž matici vnitřních parametrů označuje \mathbf{A}_1 a matice vektorů rotace a translace kamery oproti počátku \mathbf{S} je označena \mathbf{G}_1 . U matice **Proj2R** je to obdobné. Bod \mathbf{B} se bude promítat do kamerových rovin, kde se mu jednotlivě přiřadí body \mathbf{u}_1 a \mathbf{u}_2 dle rovnice

$$\mathbf{u}_1 = (u_1, v_1, 1)^T = \mathbf{A}_1 \mathbf{P}_0 \mathbf{G}_1 \mathbf{B},$$

$$\mathbf{u}_1 = \begin{pmatrix} a_u & 0 & u_0 \\ 0 & a_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (4.2)$$

Jak je znázorněno na obrázku 4.1, tak pomyslné přímky promítají 3D bod \mathbf{B} do 2D bodů \mathbf{u}_1 a \mathbf{u}_2 . Pomocí těchto přímek hledá triangulace průsečík bodů \mathbf{u}_1 a \mathbf{u}_2 . Šum a nedokonalost soustavy může způsobit to, že se pomyslné přímky vůbec neprotínají, a proto je preferován neoptimálnější výsledek s nejmenší chybou. [7]

Body \mathbf{u}_1 a \mathbf{u}_2 lze rozepsat jako $\mathbf{u}_1 = w_1(u_1, v_1, 1)$ a $\mathbf{u}_2 = w_2(u_2, v_2, 1)$, kde w označuje neznámý faktor. S rovnicí 4.2 dostaneme následující rovnice

$$w_1 u_1 = \mathbf{p1}_1^T \mathbf{B}, w_1 v_1 = \mathbf{p1}_2^T \mathbf{B}, w_1 = \mathbf{p1}_3^T \mathbf{B}, \quad (4.3)$$

$$w_2 u_2 = \mathbf{p2}_1^T \mathbf{B}, w_2 v_2 = \mathbf{p2}_2^T \mathbf{B}, w_2 = \mathbf{p2}_3^T \mathbf{B}, \quad (4.4)$$

kde $\mathbf{p1}_i$ a $\mathbf{p2}_i$ jsou i -tými řádkovými vektory z projekčních matic **Proj1L** a **Proj2R**. Po dosazení vznikne soustava čtyř rovnic o třech neznámých

$$\mathbf{p1}_3^T \mathbf{B} u_1 = \mathbf{p1}_1^T \mathbf{B}, \quad (4.5)$$

$$\mathbf{p1}_3^T \mathbf{B} v_1 = \mathbf{p1}_2^T \mathbf{B}, \quad (4.6)$$

$$\mathbf{p2}_3^T \mathbf{B} u_2 = \mathbf{p2}_1^T \mathbf{B}, \quad (4.7)$$

$$\mathbf{p2}_3^T \mathbf{B} v_2 = \mathbf{p2}_2^T \mathbf{B}. \quad (4.8)$$

Při přesunutí všech parametrů na levou stranu a následném vytknutí \mathbf{B} dostaneme maticový tvar

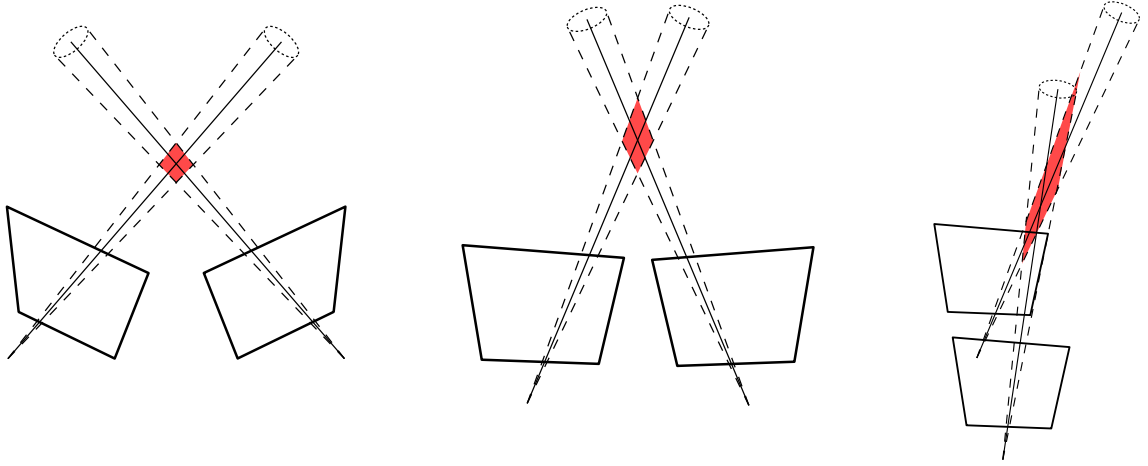
$$\mathbf{D}\mathbf{B} = \mathbf{0}, \quad (4.9)$$

kde matice \mathbf{D} , která vznikla po úpravě rovnic, obsahuje známé parametry $\mathbf{p1}_i$, $\mathbf{p2}_i$, u_1 , v_1 , u_2 , v_2 .

V této fázi stačí najít hodnoty vektoru $\mathbf{B} = (x, y, z)$ tak, aby byla splněna rovnost rovnice, kde vektor \mathbf{B} je nenulový. [7]

4.2 Pravděpodobnost rozložení odhadovaného 3D bodu

Rozložení rekonstruovaného bodu při různých úhlech paprsků znázorňuje obrázek 4.2. Za pravidlo můžeme považovat, že úhel mezi pomyslnými paprsky, které mají daným bodem procházet, určuje přesnost rekonstrukce bodu. Jedná se o mnohem lepší postup ve srovnání s více používanými jednoduššími postupy. [7]



Obr. 4.2: Nejistota rekonstrukce bodu.

Obecně pravděpodobnost určitého 3D bodu \mathbf{X} závisí na pravděpodobnosti získání jeho obrazu v každém pohledu. Pro představu si vezmeme příklad, kde je cílem odhadnout pravděpodobnost. Bod na rovině má polohu $\mathbf{X} = (X, Y)^T$ vzhledem k jeho obrazům $x = f(\mathbf{X})$ a $x' = f'(\mathbf{X})$ v obou kamerách. Předpokládáme, že měřený obrazový bod se nachází u prvního obrázku na x a samotné měření je poškozeno Gaussovým šumem s průměrnou hodnotou nula a rozptylem σ^2 , pak pravděpodobnost získání x , s ohledem na to, že skutečný obrazový bod je $f(\mathbf{X})$, je dána

$$p(x|\mathbf{X}) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left(-\frac{|f(\mathbf{X}) - x|^2}{2\sigma^2}\right), \quad (4.10)$$

se stejným významem pro $p(x'|\mathbf{X})$. Výpočet empirické distribuce:

$$p(\mathbf{X}|x, x') = \frac{p(x, x'|\mathbf{X})p(\mathbf{X})}{p(x, x')}. \quad (4.11)$$

Za předpokladu stejné předchozí pravděpodobnosti $p(\mathbf{X})$ a nezávislých obrazových měření ve dvou obrazech platí

$$p(\mathbf{X}|x, x') \sim p(x, x'|\mathbf{X}) = p(x|\mathbf{X})p(x'|\mathbf{X}). \quad (4.12)$$

Tato kapitola vychází z knihy [7].

5 NEJISTOTY MĚŘENÍ

O nejistotě měření se dá říct, že je měřítkem kvality výsledku. Neboli vyjadřuje interval hodnot, ve kterém se předpokládá, že leží pravá hodnota. Měření ovlivňuje nesčetné množství nežádoucích faktorů, které způsobí chybu v naměřené hodnotě. Chybou je myšlen rozdíl (odchylka) mezi skutečnou a naměřenou hodnotou. V praxi nenajdeme absolutně přesný měřicí přístroj nebo metodu právě kvůli těmto faktorům. Prioritní v této práci bude absolutní průměrná odchylka a směrodatná odchylka.[8]

5.1 Odchylky měření

Odchylkou měření se myslí rozdíl mezi naměřenou hodnotou a hodnotou skutečnou. Ta je nahrazována hodnotou konvenčně pravou, neboť se skutečné hodnotě přibližuje natolik, že rozdíl mezi nimi je zanedbatelný. Nahrazuje se, protože skutečnou hodnotu nemůžeme z fyzikálních příčin zjistit.

Absolutní odchylku měření Δ_X určí rozdíl mezi naměřenou hodnotou a konvenčně pravou hodnotou

$$\Delta_X = X_M - X_P, \quad (5.1)$$

kde X_M je hodnota naměřená a X_P je hodnota konvenčně pravá. Pro vyhodnocování přesnosti výsledků měření se využívá právě absolutní odchylka měření.

Relativní odchylka měření δ_X se nejčastěji v praxi vyjadřuje v procentech jako

$$\delta_X = \frac{\Delta_X}{X_M} \cdot 100. \quad (5.2)$$

Pro vyhodnocování přesnosti měřicí metody se využívá právě relativní odchylka měření. [8]

Průměrná absolutní odchylka d se definuje jako průměr absolutních odchylek tedy

$$d = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|, \quad (5.3)$$

kde x_i je hodnota daného měření, n je počet měření a \bar{x} je aritmetický průměr naměřených hodnot.

Dále je potřeba brát v úvahu chybu náhodnou a chybu hrubou.

Chyba náhodná se vyskytuje zcela náhodně. K určení se vychází ze statistických metod vyplývajících z opakovaných měření. Tato chyba je reprezentována **směrodatnou odchylkou**

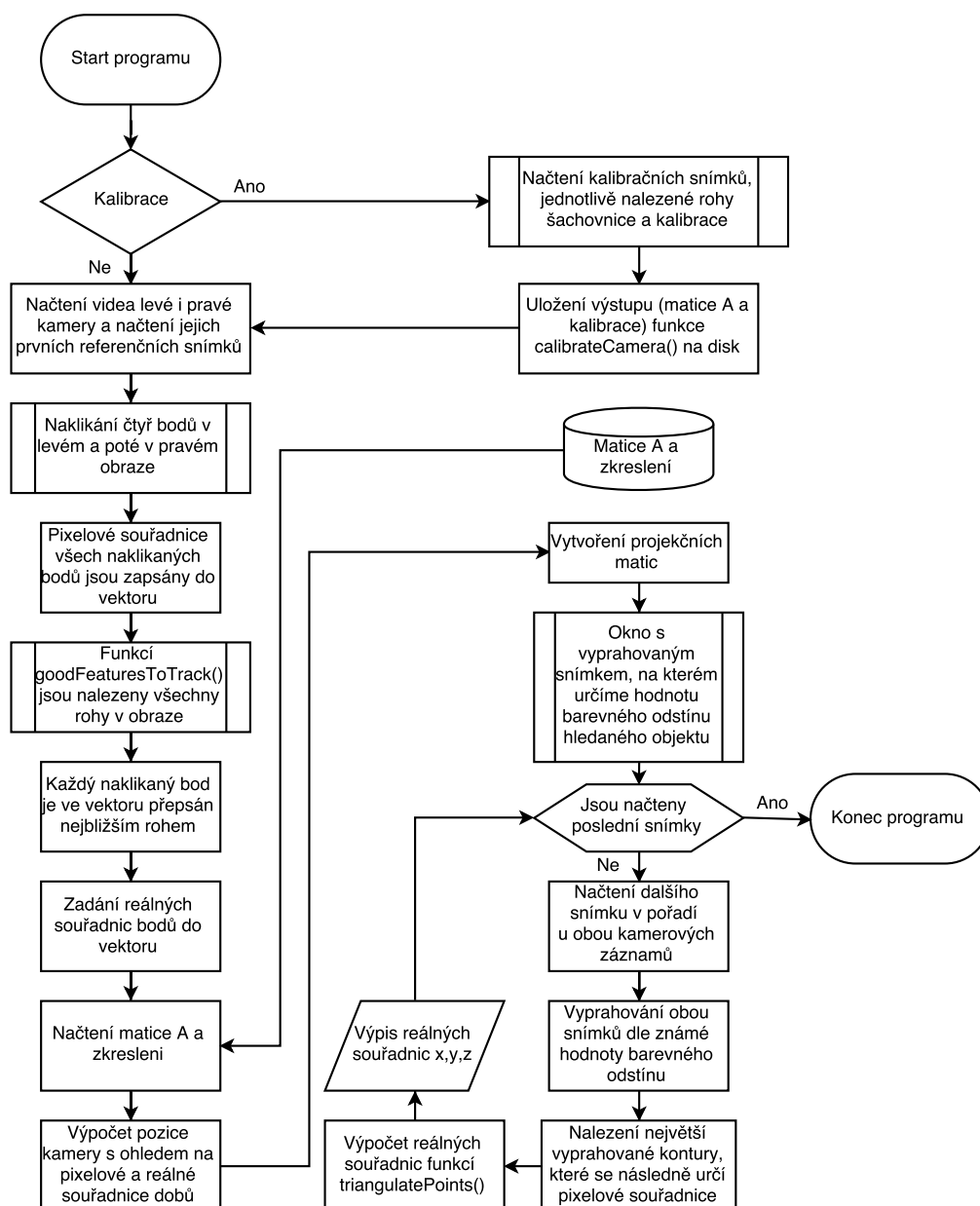
$$s_{y_i} = \sqrt{\frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n - 1}}, \quad (5.4)$$

kde y_i je hodnota daného měření, n je počet měření a \bar{y} je aritmetický průměr naměřených hodnot.

Chyby hrubé jsou typické extrémním vybočením z řady. Analyzujeme je podle velikosti a nepředvídatelnosti. Proto se s maximální a minimální hodnotou nepočítá, aby se jejich vliv eliminoval.

6 IMPLEMENTACE METOD

Na obrázku 6.1 můžeme vidět zjednodušený diagram celého programu. V této kapitole si od začátku popíšeme nejdůležitější body programu – od samotné kalibrace až po výpis výsledných informací.



Obr. 6.1: Diagram řešení.

6.1 Kalibrační snímky

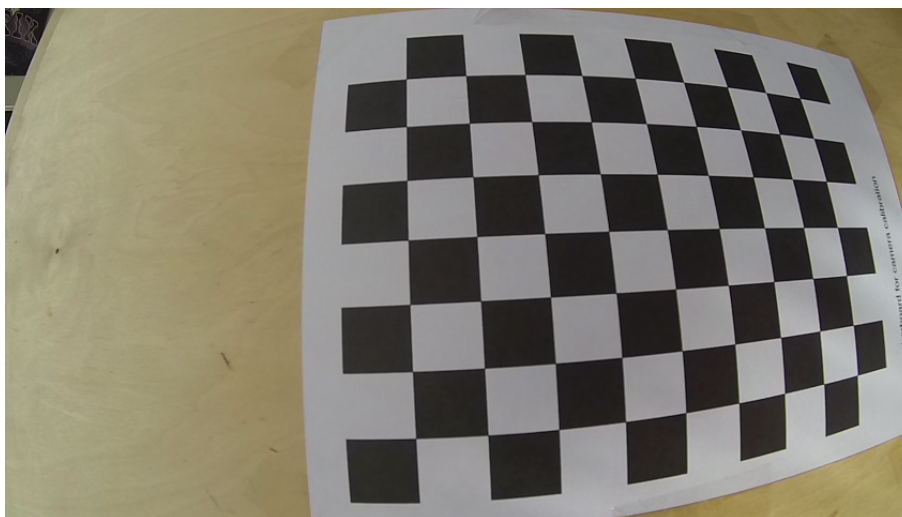
Pro zkalibrování výstupního obrazu musíme danou kamerou, v našem případě kamerou Drift, pořídit snímky, viz obrázek 6.2, na kterých se musí nacházet šachovnice s odlišnou orientací v prostoru o rozměrech 9×7 , kde je potřeba označit místa křížení (korespondenční body). V našem případě je vyhledávání korespondenčních bodů automatické, díky funkci

```
bool findChessboardCorners(InputArray image, Size patternSize,  
OutputArray corners, int flags=CALIB_CB_ADAPTIVE_THRESH+CALIB_CB_  
NORMALIZE_IMAGE )
```

z knihovny OpenCV, kde vstupními parametry je obraz a velikost kalibračního šachovnicového vzoru. Počet vodorovných a svislých vnitřních rohových bodů nám právě udává velikost šachovnicového vzoru. Tato funkce projde jednotlivě všechny kalibrační snímky. Pro vyšší přesnost obrazových souřadnic byla použita funkce

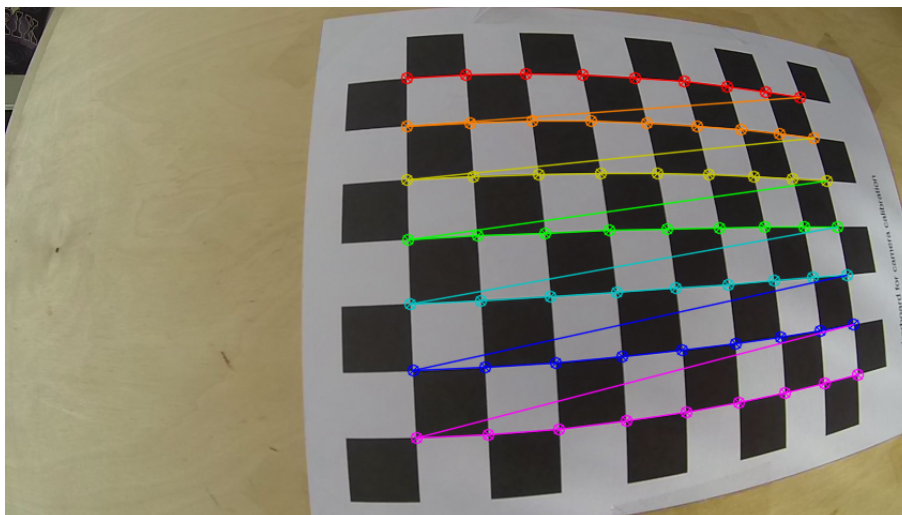
```
void cornerSubPix(InputArray image, InputOutputArray corners,  
Size winSize, Size zeroZone, TermCriteria criteria).
```

Do ní se jako vstupní parametr vloží černobílý obrázek a počáteční poloha nalezených rohů. Počáteční poloha nalezených rohů funkcí `findChessboardCorners()` se tímto přepíše na přesnější umístění rohů. [3] [9]



Obr. 6.2: Snímek šachovnice pořízený kamerou Drift určený ke kalibraci.

Pro kontrolu automaticky hledaných korespondenčních bodů byly právě tyto body vykresleny funkcí `drawChessboardCorners()`, viz obrázek 6.3. V případě, že by funkce na nějakém z obrázků nemohla body najít nebo by je našla špatně, musely by se body zadávat ručně. Z pořízených kalibračních snímků se vyberou pouze ty s nízkou kalibrační chybou, aby byla výsledná kalibrace co nejpřesnější. [11]



Obr. 6.3: Kontrolní vykreslení korespondenčních rohů.

6.2 Kalibrace kamery Drift

Jak už bylo v kapitole 3.1 řečeno, tak naše kamery mají radiální zkreslení, viz obrázek 6.4, které jsme museli odstranit z důvodu zjednodušení výpočtů a zpřesnění výsledků. V OpenCV je funkce

```
double calibrateCamera(InputArrayOfArrays objectPoints,
InputArrayOfArrays imagePoints, Size imageSize,
InputOutputArray cameraMatrix, InputOutputArray distCoeffs,
OutputArrayOfArrays rvecs, OutputArrayOfArrays tvecs, int flags=0,
TermCriteria criteria=TermCriteria( TermCriteria::COUNT+
TermCriteria::EPS, 30, DBL_EPSILON) ),
```

která nám se zadaných vstupních parametrů, jakožto vektor \mathbf{M} s prostorovými body, vektor \mathbf{m} se souřadnicemi korespondenčních bodů a rozměrů obrazu, dává výstupní matici intrinsických parametrů \mathbf{A} a koeficient zkreslení, který máme nazvaný **zkreslení**. Obě tyto informace jsou uloženy na disk do souboru *intrickaMatice.xml* a *zkresleni.xml*, abychom nemuseli provádět kalibraci při každém spuštění programu.

Na následujících obrázcích jde vidět rozdíl odstranění zkreslení na základě velikosti reprojekční chyby, kterou nám také vrací funkce `calibrateCamera()`. Reprojekční chyba je geometrickou chybou odpovídající obrazové vzdálenosti mezi projekčním a měřeným bodem. Obrázek 6.4 představuje neupravený (zkreslený) obraz kamery Drift. Obrázek 6.5 už znázorňuje odstraněné zkreslení při reprojekční chybě 7,00291, což je největší hodnota reprojekční chyby, která nám vznikla při použití špatných kalibračních snímků. Odstraněním nevyhovujících kalibračních snímků se

hodnota reprojekční chyby snížila na 1,8 (viz obrázek 6.6). A výběrem snímků s nejnižší reprojekční chybou se celková hodnota chyby snížila na 0,561422. Obrázek 6.7 tedy představuje nejlepší odstranění zkreslení, kterého bylo dosaženo. Zkreslení obrazu je odstraněno díky maticím získaným kalibrací a funkci `undistort()`, která transformuje obraz, aby kompenzovala zkreslení objektivu.[9]



Obr. 6.4: Zkreslený snímek pořízený kamerou Drift.

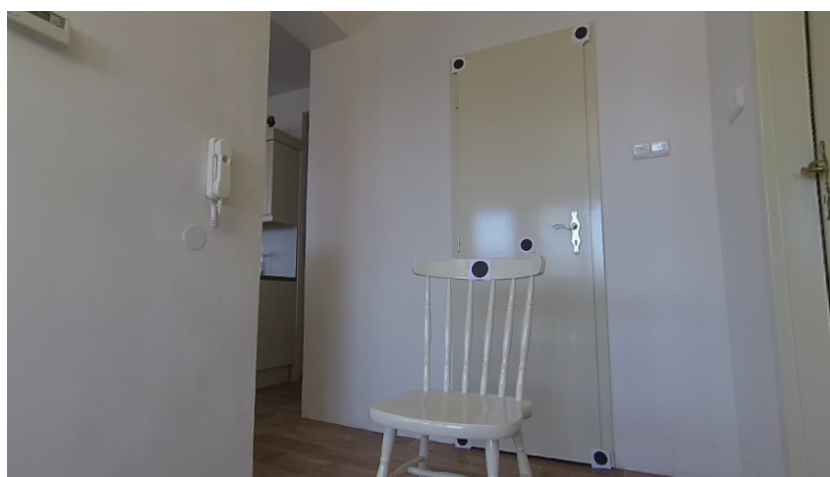


Obr. 6.5: Snímek s odstraněným zkreslením s kalibrační chybou 7,00291.

V případě změny kamery stačí vyměnit kalibrační snímky ve složce *Kalibrace* a upravit rozměry na parametry aktuálního obrazu. Po těchto úpravách by měl program správně pracovat s jakoukoliv kamerou.



Obr. 6.6: Snímek s odstraněným zkreslením s kalibrační chybou 1,8.



Obr. 6.7: Snímek s odstraněným zkreslením s kalibrační chybou 0,561422.

6.3 Pixelové souřadnice

K nalezení pixelových souřadnic používáme námi vytvořené funkce `klikMysiL()` a `klikMysiR()`, kde se při kliku myši na obraz zapíše souřadnice do vektoru `mStereo`. Po načtení prvního snímku levého i pravého záznamu se zobrazí levý obraz a spustí se jemu přiřazené funkce. Zde program čeká na naklikání čtyř bodů, které musí souhlasit s pořadím, jakým jsou zadávány světové souřadnice do vektoru `MStereo` (například když je jako první bod světových souřadnic zadán levý spodní roh zárubně $[0, 0, 0]$, tak musíme i v obraze kliknout jako první na levý roh zárubně a tak pokračujeme i pro ostatní body). Po provedení se levý obraz vymění za pravý, kde je postup obdobný. Poté se tyto body porovnávají s body nalezenými funkcí

```
void goodFeaturesToTrack(InputArray image, OutputArray corners,
```

```
int maxCorners, double qualityLevel, double minDistance,
InputArray mask=noArray(), int blockSize=3, bool
useHarrisDetector=false, double k=0.04 ).
```

Funkce `goodFeaturesToTrack()` pracuje tak, že do ní vložíme černobílý obraz, který nám s originálního RGB obrazu udělá funkce

```
void cvtColor(InputArray src, OutputArray dst, int code,
int dstCn=0 ),
```

určíme si jaké parametry rohů vyžadujeme a funkce nám dá výsledný vektor se všemi nalezenými rohy v obraze s jejich pixelovými souřadnicemi. Každému označenému bodu je přiřazen takový roh, který je porovnávánému bodu nejbližší.

6.4 Světové souřadnice

V naší práci jsme použili čtyři body ze světového souřadnicového systému, kde levý spodní roh zárubně udává počátek systému. Zbylé body se nachází v rovině (x, y) . Tyto body jsou zapisovány do vektoru **MStereo** v pořadí:

1. Bod $[0, 0, 0]$ – levý spodní roh,
2. Bod $[0, \mathbf{v}, 0]$ – levý horní roh,
3. Bod $[\mathbf{s}, \mathbf{v}, 0]$ – pravý horní roh,
4. Bod $[\mathbf{s}, 0, 0]$ – pravý spodní roh,

kde \mathbf{v} je výška a \mathbf{s} je šířka zárubně. Námi zvolená zárubeň má rozměry $v = 201\text{cm}$ a $s = 87\text{cm}$. Z toho plyne, že osa y je totožná s levou stranou zárubně a rovina (x, z) je totožná s podlahou, přičemž osa x je rovnoběžná se stěnou. Pro lepší představu jsou osy vyznačeny na obrázku 2.1.

Záleží jenom na nás, v jakém pořadí dané body zapíšeme, ale poté se musí přizpůsobit i zadávání pixelových souřadnic.

6.5 Pozice kamer vůči světovému souřadnicovému systému

Pro následující krok bylo důležité najít vektor s rotací a vektor s translací souřadnicového systému kamer oproti námi zvoleným světovým souřadnicím. K tomuto účelu jsme použili funkci

```
bool solvePnP(InputArray objectPoints, InputArray imagePoints,
InputArray cameraMatrix, InputArray distCoeffs, OutputArray rvec,
OutputArray tvec, bool useExtrinsicGuess=false, int flags=ITERATIVE),
```

která ze vstupních parametrů (vektor se světovými souřadnicemi, vektor s pixelovými souřadnicemi, matice intrinsických parametrů **A** a koeficient zkreslení) vypočítá námi hledané vektory, kde vektor rotace udává pomocí tří hodnot otočení kamery v osách x , y a z , a kde vektor translace na těchto osách udává posunutí, taktéž pomocí tří hodnot. [9]

6.6 Projekční matice

Díky získaným výstupním vektorům rotace a translace z funkce `solvePnP()` jsme schopni vytvořit projekční matice obou kamer tak, jako je tomu v rovnici 4.1. Ovšem ještě před tím musíme převést rotační vektor na rotační matici (3×3) tak, aby všechny hodnoty ve vzorci souhlasily. K tomu nám posloužila další funkce

```
void Rodrigues(InputArray src, OutputArray dst, OutputArray  
jacobian=noArray()),
```

opět z knihovny OpenCV, kde vložíme jako vstupní informaci právě náš rotační vektor a jako výstupní informaci dostaneme potřebou matici. Princip, z něhož funkce vychází, najdeme v [10]. [9]

Poté vytvoříme matici **G** (viz 3.5) a vynásobíme ji maticí intrinsických parametrů **A**. Postup, z něhož dostaneme projekční matice **Proj1L** a **Proj2R**, je stejný pro obě kamery.

6.7 Nalezení bodu v prostoru

Posledním a zároveň tím nejpodstatnějším bodem naší práce bylo zjistit prostorové souřadnice jakéhokoliv bodu v námi snímaném prostoru. Povedlo se nám to udělat až do takové fáze, kde námi vybraný objekt měřicí zařízení sleduje (na základě jeho barevného odstínu) a jeho souřadnice vypisuje do textové konzole.

6.7.1 Předpřípravení obrazu

Jako první se musí vstupní obraz RGB převést na obraz barevného modelu HSV. Více o barevných modelech v [6]. S tím nám opět pomůže funkce `cvtColor()` z balíčku knihoven OpenCV. Teď už obraz nebude tvořen třemi barevnými složkami jako R (červená), G (zelená) a B (modrá), ale budou ho tvořit složky H (barevný tón), S (sytnost barvy) a V (hodnota jasu).

V dalším kroku použijeme funkci

```
void inRange(InputArray src, InputArray lowerb, InputArray upperb,  
OutputArray dst),
```

která z HSV obrazu náš hledaný objekt vyprahuje. Jelikož je naším hledaným objektem žlutá reflexní kulička, můžeme si dovolit nastavit napevno barevný tón do rozsahu 150–255 a sytost barvy do rozsahu 100–255. Barvu už nastavujeme ručně trackbarem. Ten můžeme posunovat po škále 0–179 a zahrnuje vždy barevný rozsah o šířce 10. Po nalezení barevného rozsahu hledaného objektu si program tyto hodnoty uloží. Na vyprahovaném obrazu použijeme pro zpřesnění k dalšímu pokračování funkci

```
void blur(InputArray src, OutputArray dst, Size ksize, Point
anchor=Point(-1,-1), int borderType=BORDER_DEFAULT ),
```

která zcelí a vyhladí kontury v binárním obraze.

6.7.2 Nalezení bodu a výpis souřadnic

V tuto chvíli už program pracuje samostatně. Postupně snímek po snímku projde celým videem a souřadnice objektu vypisuje do konzole.

Program současně vezme po snímku z každé kamery. Ty jsou převedeny pomocí funkce `cvtColor()` na barevný model HSV. Poté se snímky vyprahují funkcí `inRange()`, která má hodnoty pro zpracování uložené v paměti, viz kapitola 6.7.1. Pro zpřesnění se vyprahovaným konturám zjemní hrany funkcí `blur()`. Pro nalezení kontury patřící našemu objektu byl zvolený postup takový, že funkcí

```
void findContours(InputOutputArray image, OutputArrayOfArrays
contours, OutputArray hierarchy, int mode, int method, Point
offset=Point())
```

byly nalezeny všechny kontury v obrazech a vybrána byla ta, která měla největší velikost. Funkce

```
void rectangle(Mat& img, Rect rec, const Scalar& color,
int thickness=1, int lineType=8, int shift=0 )
```

ohraničí obdélníkem konturu a pixelové souřadnice jeho středu se použijí jako obrazové umístění objektu.

K finálnímu určení reálných souřadnic nám posloužila funkce

```
void triangulatePoints (InputArray projMatr1, InputArray projMatr2,
InputArray projPoints1, InputArray projPoints2, OutputArray points4D)
```

z balíčku OpenCV. Do ní se vloží jako vstupní informace projekční matice obou kamer (v našem případě **Proj1L** a **Proj2R**) a pixelové souřadnice hledaného objektu, které byli automaticky nalezeny (popsáno výše). Výstupní informací jsou požadované prostorové souřadnice v homogenním tvaru. Abychom však dostali souřadnice $[x, y, z]$ tak, jak je známe, musíme jednotlivé souřadnice vydělit souřadnicí poslední. [9]

Funkce je založena právě na principu lineární triangulace popsané v kapitole 4.1.

6.8 Popis chodu programu

Po spuštění programu se objeví dotaz „`Chcete spustit kalibraci? a/n`“, kterým si vybereme, zda chceme z uložených obrázků ve složce *Kalibrace* vypočítat matici vnitřních parametrů kamery **A** a koeficient zkreslení, jenž budou následně uloženy do souborů *.xml* (možnost **a**), nebo zda program bude pokračovat se soubory *.xml*, do kterých byly nahrány informace z poslední provedené kalibrace (možnost **n**). Toto opatření bylo provedeno ze dvou důvodů. Prvním důvodem je časově delší náročnost výpočtů a druhým důvodem (důležitějším) je to, aby se nám neustále nepřepisoval prostor na disku. Kalibrace mohla být samozřejmě provedena jenom jednou a pak už se mohlo vycházet z uložených parametrů, ale takto je program připravený na případnou změnu kamer tzn. výpočet nových hodnot matice **A** a koeficientu zkreslení.

Po předchozím kroku se objeví snímek levé kamery čekající na naklikání čtyř bodů, které míříme na rohy zárubně dotýkajících se stěny. Pořadí naklikání bodů určuje, jak už bylo výše zmíněno, pořadí světových souřadnic tzn. jako první označíme levý spodní roh, poté levý horní roh, jako třetí v pořadí pravý horní roh a nakonec pravý spodní roh. Po odklikání všech čtyř rohů se snímek zavře a objeví se snímek pravé kamery, kde je postup stejný. Okno se snímkem pravé kamery se taky zavře a ukážou se nám pro kontrolu okna s vyznačenými rohy, které byli nejbližší naklikaným bodům.

Jestliže jsou rohy správně označeny, tak program může pokračovat dál. Zobrazí se dvě okna. V jednom je snímek z kamery bez úpravy (pro kontrolu zda se pracuje správný objekt na správném místě) a v druhém okně se nám vyskytuje slider a vyprahovaný obraz. Sliderem vybereme rozsah (od x do $x + 10$) odstínu barvy jakou má náš hledaný objekt. To poznáme tak, že na pozici objektu se v černém snímku objeví bílá kontura. Poté, co objektu určíme jeho barevný odstín a stiskneme klávesu enter, program už pracuje sám. Zobrazí okno, ve kterém se snímky mění po 5 ms a zaokrouhlené souřadnice objektu, který je hledán podle škály barevného odstínu, se vypisují v levém horním rohu okna zaokrouhleny na celá čísla,

viz obrázek 6.8. Když objekt není na snímku nalezen (není v obraze nebo je nějaký jiný objekt mezi hledaným objektem a kamerou) vypisují se souřadnice objektu, který byl na obou kamerách viděn naposledy. Posledním snímkem videí se program ukončí. Nezaokrouhlení souřadnice jsou vypisovány do konzole.



Obr. 6.8: Screenshot okna se souřadnicemi objektu v daném snímku.

7 ZHODNOCENÍ VÝSTUPNÍCH HODNOT

K finálnímu měření jsme si vybrali scénu, která je popsána v kapitole 2.1. Provedli jsme tři měření, která se lišila pouze rozestavěním kamer. V prvním měření kamery svíraly takřka nulový úhel, jelikož scénu snímaly postaveny vedle sebe, viz fotografie 7.1. Ve druhém měření od sebe byly kamery vzdáleny 160 cm, což mezi nimi svíralo úhel přibližně 40° , viz fotografie 7.2. Ve třetím jakožto posledním měření byly od sebe kamery vzdáleny nejdále. Rozestup mezi kamerami byl 405 cm. Tímto kamery mezi sebou svíraly úhel o hodnotě 85° , viz fotografie 7.3. Kamery byly posunovány po přímce, která byla rovnoběžná se snímanou stěnou, tudíž kamera vždy stála ve vzdálenosti 220 cm od snímané stěny. Každé měření zachytilo souřadnice objektu na 27 pozicích, které byly pro každé měření stejné. Tyto pozice se nacházejí v pomyslném kvádru před dveřmi, kde jeho spodní podstava má reálné souřadnice rohů $A = [-30; 108; 82, 5]$ cm, $B = [131, 5; 108; 82, 5]$ cm, $C = [131, 5; 108; 82, 5]$ cm a $D = [-30; 108; 21, 5]$ cm a horní podstava má reálné souřadnice rohů $E = [-30; 178; 82, 5]$ cm, $F = [131, 5; 178; 82, 5]$ cm, $G = [131, 5; 178; 21, 5]$ cm a $H = [-30; 178; 21, 5]$ cm, viz obrázek 7.4 pro představu.



Obr. 7.1: Fotografie prvního měření (kamery svírají nulový úhel).

V obrázku je také každá pozice označena číslem, které zároveň udává v jakém pořadí byly tyto pozice měřeny.

Měřeným objektem byla žlutá reflexní kulička, která byla umístěna na stojánku. Stojánek byl použit pro dobrou manipulaci při přemísťování v osách x a z a také proto, že se měřenému objektu (kuličce) dala přesně nastavit hodnota v ose y .

Reálné souřadnice pozic měřeného objektu jsou zapsány v tabulce 7.1, které budeme porovnávat se souřadnicemi vypočítanými naším programem. A v tabulce 7.5



Obr. 7.2: Fotografie druhého měření (kamery svírají úhel 40°).



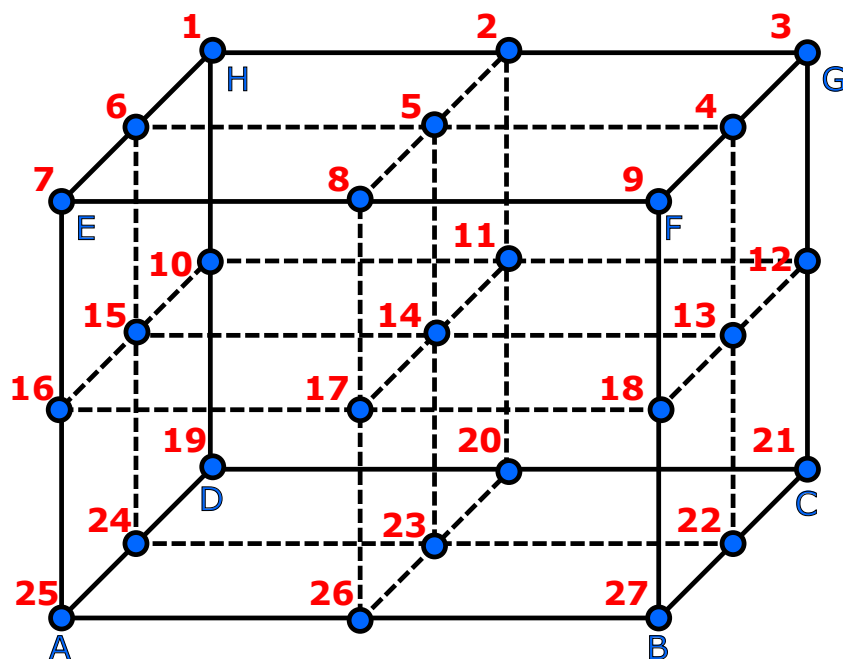
Obr. 7.3: Fotografie třetího měření (kamery svírají úhel 85°).

najdeme vypočtené hodnoty absolutní průměrné odchylky, směrodatné odchylky, průměrné odchylky v relativní míře a relativní chyby všech souřadnic.

7.1 První měření – kamery svírající nulový úhel

Hodnoty tohoto měření jsou zaznamenány v tabulce 7.2. Z hodnot jsme vypočetli, že

- v ose x byla absolutní průměrná odchylka 2,3 cm se směrodatnou odchylkou o hodnotě 2,59 cm,
- v ose y byla absolutní průměrná odchylka 4,68 cm se směrodatnou odchylkou o hodnotě 6,1 cm,
- v ose z byla absolutní průměrná odchylka 9,23 cm se směrodatnou odchylkou



Obr. 7.4: Prostor ve kterém měření probíhalo a číselně označené pozice.

o hodnotě 11,22 cm.

Je zřejmé, že měření v ose z má podstatně větší odchylku než měření u zbývajících os. Způsobují to hodnoty pozic, které dosahují téměř ke 40 cm odchylky v absolutní míře. V 16 pozicích přesáhla velikost odchylky 10 cm v ose z v ose y těchto hodnot bylo 6 a v ose x ani jedna z hodnot tuto hranici nepřesáhla. Z toho plyne, že pro toto rozestavení kamer bylo měření nejpřesnější u souřadnic X .

Celková absolutní průměrná odchylka činila 9,29 cm s celkovou směrodatnou odchylkou 11,62 cm. Měření s kamerami v těchto pozicích se dá považovat za nepříliš přesné a přesvědčivé.

7.2 Druhé měření – kamery svírající úhel 40°

Ve druhém měření kamery svíraly úhel 40° s rozestupem 160 cm mezi sebou. Reflexní kulička byla měřena ve stejném pořadí na úplně stejných pozicích jako u předchozího měření tak, jako je znázorněno na obrázku 7.4. V tabulce 7.3 najdeme naměřené hodnoty tohoto měření. Z naměřených a reálných hodnot byla opět vypočtena absolutní průměrná odchylka se směrodatnou odchylkou jak u všech os tak i u celkového výsledku. Hodnoty vyšly tak, že

- v ose x byla odchylka 1,71 cm se směrodatnou odchylkou 2,03 cm,
- v ose y byla odchylka 1,98 cm se směrodatnou odchylkou 2,44 cm,
- v ose z byla odchylka 3,29 cm se směrodatnou odchylkou 3,94 cm.

Tab. 7.1: Reálné hodnoty měřeného objektu v daných pozicích.

Pozice	X [cm]	Y [cm]	Z [cm]	Pozice	X [cm]	Y [cm]	Z [cm]
1	-30	178	21,5	15	-30	138	42
2	51	178	21,5	16	-30	138	82,5
3	131,5	178	21,5	17	51	138	82,5
4	131,5	178	42	18	131,5	138	82,5
5	51	178	42	19	-30	108	21,5
6	-30	178	42	20	51	108	21,5
7	-30	178	82,5	21	131,5	108	21,5
8	51	178	82,5	22	131,5	108	42
9	131,5	178	82,5	23	51	108	42
10	-30	138	21,5	24	-30	108	42
11	51	138	21,5	25	-30	108	82,5
12	131,5	138	21,5	26	51	108	82,5
13	131,5	138	42	27	131,5	108	82,5
14	51	138	42	-	-	-	-

Tab. 7.2: Naměřené souřadnice objektu v daných pozicích v prvním měření.

Pozice	X [cm]	Y [cm]	Z [cm]	Pozice	X [cm]	Y [cm]	Z [cm]
1	-22	178,35	15,83	15	-23,36	145,41	27,87
2	50,15	179,82	17,79	16	-35,03	157,16	43,54
3	124,25	176,78	21,6	17	54,23	162,82	43,19
4	123,58	174,91	38,42	18	135,22	150,14	57,89
5	50,52	184,92	30,7	19	-24,99	110,15	11,19
6	-25,13	184,69	25,1	20	51,46	109,88	16,42
7	-27,81	193,79	49,51	21	125,35	108,3	24,48
8	52,19	193,62	59,61	22	126,5	110,06	39,77
9	129,53	186,23	60,31	23	50,69	111,11	37,93
10	-23,52	140,83	15,38	24	-26,8	111,47	26,54
11	50,11	144,05	12,92	25	-32,01	117,09	53,48
12	125,06	138,51	22,14	26	50,9	116,25	66,75
13	128,51	142,2	33,8	27	131,22	114,83	66,74
14	51,66	149,25	22,18	-	-	-	-

Můžeme si všimnout, že se hodnoty výrazně oproti předchozímu měření snížily a nevyskytují se zde ani tak hrubé chyby. Ale osa z má pořád znatelně větší průměrnou odchylku se směrodatnou odchylkou než zbývající osy. Opět tomu napomohlo, že v pěti případech přesáhla odchylka v absolutní míře hodnotu 10 cm. V ose y se vyskytl pouze jeden případ a v ose x se žádný takový neobjevil.

Celková absolutní průměrná odchylka činila 3,15 cm s celkovou směrodatnou odchylkou o hodnotě 3,85 cm. Z čehož v porovnání s předchozím měřením vyplývá, že celková odchylka i směrodatná odchylka se přibližně třikrát snížily.

Tab. 7.3: Naměřené souřadnice objektu v daných pozicích ve druhém měření.

Pozice	X [cm]	Y [cm]	Z [cm]	Pozice	X [cm]	Y [cm]	Z [cm]
1	-25,11	176,16	24,35	15	-27,73	138,93	38,48
2	50,33	180,39	20,02	16	-30,66	143,34	69,6
3	124,03	173,07	28,74	17	54,12	145,38	70,91
4	125,3	175,01	44,4	18	131,98	139,77	76,62
5	50,73	183,28	36,48	19	-25,61	107,96	22,58
6	-36,63	178,4	40,4	20	52,1	108,5	16,7
7	-28,61	183,89	70,68	21	127,36	106,42	27,14
8	51,74	188,16	70,33	22	127,96	107,78	43,71
9	127,75	178,07	76,22	23	52,78	109,99	34,59
10	-25,24	136,46	23,23	24	-27,66	109,32	38,41
11	51,21	139,14	18,54	25	-31,21	112,98	69,29
12	126,4	134,21	28,95	26	53,99	112,97	69,73
13	128,57	136,39	44,01	27	133,77	110,47	75,18
14	52,18	140,99	34,77	-	-	-	-

7.3 Třetí měření – kamery svírající úhel 85°

V posledním měření byly od sebe kamery vzdáleny nejdále a to 405 cm a také svíraly největší úhel 85°. Hodnoty měření byly opět zaznamenány viz tabulka 7.4. Opět se měřila reflexní kulička na úplně stejných pozicích a ve stejném pořadí. Výpočty říkají, že

- v ose x byla absolutní průměrná odchylka 1,46 cm se směrodatnou odchylkou o hodnotě 1,81 cm,
- v ose y byla absolutní průměrná odchylka 0,81 cm se směrodatnou odchylkou o hodnotě 1,11 cm,
- v ose z měla tato odchylka hodnotu 1,73 cm se směrodatnou odchylkou o hodnotě 2,12 cm.

Při pohledu na vypočtené hodnoty vidíme, že odchylky jsou v porovnání s předcházejícími měřeními nejmenší. V tomto měření se nevyskytla žádná hrubá chyba. Dokonce žádná souřadnice nepřekonala chybu o hodnotě 10 cm.

Celková absolutní průměrná odchylka činila 1,78 cm s celkovou směrodatnou odchylkou 2,11 cm. Tímto se potvrdilo očekávání, že s rostoucím úhlem kamerami sevřeným roste i přesnost měření. Vyplývá to z teorie, viz kapitola 4.2, kde se mají dva pomyslné paprsky vycházející z kamer protnout v měřeném objektu. Z třetího měření můžeme konstatovat, že kamery v tomto rozpoložení zapříčiní nejpřesnějších výsledků.

Tab. 7.4: Naměřené souřadnice objektu v daných pozicích ve třetím měření.

Pozice	X [cm]	Y [cm]	Z [cm]	Pozice	X [cm]	Y [cm]	Z [cm]
1	-31,05	176,54	24,65	15	-32,73	136,17	40,66
2	48,14	177,39	24,64	16	-35,51	137,2	78,73
3	126,9	173,01	27,92	17	52,03	137,38	76,66
4	128,59	174,52	45,73	18	135,13	136,23	77,52
5	49,17	178,58	41,49	19	-32,77	106,29	19,94
6	-33,08	177,29	43,91	20	50,14	106,3	19,67
7	-35,67	178,69	80,41	21	129,06	106,16	23,28
8	48,32	180,35	78,8	22	131,33	105,18	41,26
9	130,19	176,24	79,7	23	49,84	106,56	37,7
10	-31,53	135,91	22,4	24	-34,59	106,59	39,35
11	51,15	136,06	21,73	25	-37,15	106,65	76,31
12	130,25	133,73	25,64	26	50,75	106,95	74,35
13	132,28	134,27	43	27	134,41	106,8	75,02
14	51,42	136,26	39,7	-	-	-	-

Tab. 7.5: Vypočtené hodnoty.

Měření	Absolutní průměrná odchylka [cm]			
	X	Y	Z	Celková
1.	2,30	4,68	9,23	9,29
2.	1,71	1,98	3,29	3,16
3.	1,46	0,81	1,73	1,78
Měření	Směrodatná odchylka [cm]			
	X	Y	Z	Celková
1.	2,59	6,10	11,22	11,62
2.	2,03	2,44	3,94	3,85
3.	1,81	1,11	2,12	2,11
Měření	Průměrná odchylka v relativné míře [%]			
	X	Y	Z	
1.	7,13	4,89	26,72	
2.	6,19	1,91	17,96	
3.	5,70	1,35	7,60	
Měření	Relativní chyba [%]			
	X	Y	Z	
1.	1,42±1,60	6,69±8,71	15,14±18,39	
2.	1,06±1,26	2,83±3,48	5,40±6,46	
3.	0,90±1,12	1,15±1,59	2,84±3,48	

8 ZÁVĚR

V rámci bakalářské práce byly prostudovány teoretické principy kalibrace kamer a postupy pro získání prostorové informace z digitálního obrazu. Za pomoci těchto znalostí bylo vytvořeno jednoduché měřicí zařízení, které dokáže s určitou přesností určit 3D souřadnice jakéhokoliv bodu nacházejícího se ve snímaném prostoru. Zařízení je složeno ze dvou sportovních kamer Drift a námi vytvořeného softwaru, který vypočítá trojrozměrné souřadnice X , Y a Z , využívajícího metodu triangulace.

Zařízení bylo testováno v rámci tří měření. Měřeným objektem byla žlutá reflexní kulička. Podle barevného odstínu si program sám v obraze kuličku najde a vypočítá její reálné souřadnice. Kulička byla při každém měření 27krát změněna pozice v prostoru a její souřadnice v dané pozici byly odečteny z výpisu v konzole. Umístění kuličky v prostoru bylo vždy pro všechna měření stejné. Měření se lišila pouze rozestavěním kamer a úhlem mezi sebou svírajícím.

Měření ukázala, že přesnost výsledků roste s úhlem, který kamery svírají. Neboť měření s nulovým úhlem mělo přibližně 5krát větší celkovou směrodatnou odchylku i celkovou absolutní průměrnou odchylku v porovnání s měřením, kde kamery svíraly úhel 85° . Nejmenší celkové absolutní průměrné odchylky o hodnotě 1,78 cm a nejmenší celkové směrodatné odchylky o hodnotě 23,11 cm bylo u zařízení docíleno tedy ve třetím měření, kdy kamery svíraly úhel 85° . Kulička byla přemísťována v prostoru o rozměrech $161,5 \times 70 \times 61$ cm.

Výsledné zařízení by se mělo používat s rozestavěním kamer, jako tomu bylo u měření třetího. Docílí se tak nejpřesnějších výsledků. Za předpokladu, že se ve scéně neobjeví žádný objekt o stejném barevném odstínu a scéna bude po celou dobu měření stejně osvětlena, by měl program pracovat bez výrazných chyb.

Zařízení by mohlo být využito například pro míčové hry, u kterých by sledovalo polohu míče (fotbal) či míčku (tenis, stolní tenis) a určovalo by, zda míč (míček) opustil či neopustil určitou zónu. Dalším příkladem využití by mohlo být střežení určitého objektu, který nesmí opustit určitý prostor (zabezpečení sošky v muzeu). Předpokládáme, že ke správné funkčnosti by však musely být provedeny úpravy samotného programu, zvýšena snímková rychlost a taky zvětšeno rozlišení snímků.

LITERATURA

- [1] JOHNSON, B. *Professional visual studio 2013: selected readings*. 1st edition. Washington: Microsoft Press. Penguin education. ISBN 11-188-3204-3.
- [2] About. *OpenCV* [online]. Itseez, c2016 [cit. 20.11.2016]. Dostupné z URL: <<http://opencv.org/about.html>>.
- [3] BRADSKI, Gary R. *Learning OpenCV*. Sebastopol: O'Reilly, c2008. ISBN 9780596516130.
- [4] DRIFT. *Drift Ghost-S návod k použití*.
- [5] ŘÍHA, Kamil. *Pokročilé techniky zpracování obrazu* [online]. Vysoké učení technické v Brně Fakulta elektrotechniky a komunikačních technologií Ústav telekomunikací Technická 12, 616 00 Brno, 2012 [cit. 20.11.2016]. 143s. ISBN 978-80-214-4894-0. Dostupné z URL: <moodle.vutbr.cz>.
- [6] HORÁK, K., KALOVÁ, I., PETYOVSKÝ, P., RICHTER, M. *Počítačové vidění*. Brno: VUT Fakulta elektrotechniky a komunikačních technologií, 2008. 132 s.
- [7] HARTLEY, Richard. a Andrew. ZISSERMAN. *Multiple view geometry in computer vision*. 2nd ed. New York: Cambridge University Press, 2003. ISBN 0521540518.
- [8] BARTUŠEK, Karel. *Měření v elektrotechnice*. 2., přeprac. a dopl. vyd. Brno: VUTIUM, 2010. ISBN 9788021441606.
- [9] Camera Calibration and 3D Reconstruction. *OpenCV* [online]. opencv dev team, c2001-2014, Last updated on Dec 08, 2016 [cit. 8.11.2016]. Dostupné z URL: <http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html>.
- [10] SLABAUGH, Gregory G. Computing Euler angles from a rotation matrix. *Department of Computer Science* [online]. 2015 [cit. 2016-11-08]. Dostupné z URL: <<http://www.staff.city.ac.uk/~sbbh653/publications/euler.pdf>>.
- [11] KARÁSEK, Miroslav. *Zpracování a vizualizace stereo snímků*. Fakulta elektrotechniky a komunikačních technologií VUT v Brně, 2012. Diplomová práce. Fakulta elektrotechniky a komunikačních technologií VUT v Brně. Vedoucí práce Petr Petyovský.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CMOS	technologie Complementary Metal — Oxide–Semiconductor používána na výrobu senzorů a čipů
FPS	Frame Per Second (snímek za sekundu)
HSV	Hue, Saturation, Value (barevný tón, sytost barvy, hodnota jasu) – barevný model
OpenCV	Open Source Computer Vision – Otevřený software počítačového vidění
RGB	Red, Green, Blue (červená, zelená, modrá) – barevný model
WVGA	Wide Video Graphics Array - standard počítačových zobrazovacích zařízení s rozlišením 640x480
3D	Trojrozměrný

SEZNAM PŘÍLOH

A Obsah přiloženého CD

43

A OBSAH PŘILOŽENÉHO CD

V kořenovém adresáři přiloženého CD najdeme PDF soubor s vypracovanou bakalářskou prací a složku Program, ve které se nachází soubory pro spuštění vytvořeného programu.

```
/ ..... kořenový adresář přiloženého CD
├── BP_Slama.pdf ..... bakalářská práce
├── Program
│   ├── BP_Slama
│   │   ├── BP_Slama.sln
│   │   ├── BP_Slama.VC.db
│   │   └── BP_Slama
│   │       ├── intrickaMatice.xml .Soubor s uloženou hodnotou intrické matice
│   │       ├── zkresleni.xml ..... Soubor s uloženou hodnotou zkresleni
│   │       ├── Source.cpp ..... Hlavní běh programu
│   │       ├── Funkce.h ..... Vytvořené funkce programu
│   │       ├── BP_Slama.vcxproj
│   │       ├── BP_Slama.vcxproj.filters
│   │       └── Kalibrace ..... Složka s 25 kalibračními snímky
│   │           ├── Kalibrace01.png
│   │           ├── Kalibrace02.png
│   │           ├── Kalibrace03.png
│   │           ├── Kalibrace04.png
│   │           ├── Kalibrace05.png
│   │           ├── Kalibrace06.png
│   │           ├── Kalibrace07.png
│   │           ├── Kalibrace08.png
│   │           ├── Kalibrace09.png
│   │           ├── Kalibrace10.png
│   │           ├── Kalibrace11.png
│   │           ├── Kalibrace12.png
│   │           ├── Kalibrace13.png
│   │           ├── Kalibrace14.png
│   │           ├── Kalibrace15.png
│   │           ├── Kalibrace16.png
│   │           ├── Kalibrace17.png
│   │           ├── Kalibrace18.png
│   │           ├── Kalibrace19.png
│   │           ├── Kalibrace20.png
│   │           ├── Kalibrace21.png
│   │           ├── Kalibrace22.png
│   │           ├── Kalibrace23.png
│   │           ├── Kalibrace24.png
│   │           └── Kalibrace25.png
│   └── Video ..... Záznamy z obou kamer rozděleny do složek
│       └── leva
```

```
└─ Video_leva.....Videa pořízená levou kamerou
    └─ Video (1).mp4
    └─ Video (2).mp4
    └─ Video (3).mp4
└─ prava
    └─ Video_prava .....Videa pořízená pravou kamerou
        └─ Video (1).mp4
        └─ Video (2).mp4
        └─ Video (3).mp4
```